Bandwidth-efficient graphics with ARM Mali GPUs

June 27th (Friday), 2014
Hessed Choi @ ARM
Memory Bandwidth
Bandwidth
Where does it go?

- Vertex load
- Varyings
- Textures
- Framebuffer output
Bandwidth
Where does it go?

- Vertex load
- Varyings
- Textures
- Framebuffer output
Bandwidth

Where are we?

- Less power = less memory bandwidth

  - Desktop: 170 Watts to >300 Watts… That’s just the GPU!

  - Console: 80 - 100 Watts (CPU/GPU/WiFi/Network)

  - Mobile: 3 - 7 Watts (CPU/GPU/Modem/WiFi)

- Need smarter solutions
Mali Architecture
Mali is a tile-based deferred rendering architecture

- Framebuffer is divided into tiles
- Renders tile by tile
- 16x16 tile size
  - Color
  - Depth
  - Stencil
Deferred Shading and Extensions Support
Deferred Shading

Limitation

- Popular technique on PC and console games
- Very memory bandwidth intensive
- Traditionally not a good fit for mobile
Extensions (1)

Shader Framebuffer Fetch

- Fragment shader extensions for OpenGL® ES 2.0 and above
- Allows reading of existing framebuffer color, depth and stencil values
- Enables:
  - Programmable blending
  - Programmable depth/stencil testing
  - Soft particles
  - Reconstruction of 3D position
  - etc

http://www.khronos.org/registry/gles/extensions/ARM/ARM_shader_framebuffer_fetch.txt
http://www.khronos.org/registry/gles/extensions/ARM/ARM_shader_framebuffer_fetch_depth_stencil.txt
Extensions (2)

Shader Pixel Local Storage (PLS)

- Fragment shader extension for OpenGL® ES 3.0 and above

- On the ARM® Mali™-T600 series this amounts to 128-bits per pixel
  - Mali-T760 can support even more data per pixel

- Enables reading and writing the current pixel’s data that is persistent throughout the lifetime of the framebuffer

- Independent of framebuffer format

http://www.khronos.org/registry/gles/extensions/EXT/EXT_shader_pixel_local_storage.txt
Deferred Shading

Resolve

- Compute final pixel color based on the Pixel Local Storage data
- Output to current framebuffer format
Deferred Shading
Bandwidth Comparison

Write MB/s
- Using extensions
- Multiple render targets

Read MB/s

Total MB/s

0. 750. 1500. 2250.

Using extensions
Multiple render targets

defered shading example
rendering to 4xRGBA8 1080p@30fps
Future

- Various deferred shading/lighting
- Order independent transparency
- Deferred virtual texturing
- Volume rendering
- etc, etc, etc

Questions?

Thank you.